

OS - Homework 2

Connor Finley

2017/10/21

1

A system has a total of 8 resources of a particular type. There are 4 processes running with their maximum resources needs and current allocation of resources as shown below:

Process	Current Allocation	Max. Need
P1	1	3
P2	1	2
P3	2	5
P4	2	6

a.

Specify all the safe sequence from the state shown above.

$m = 8$

$n = 4$

Resources left over ($m - \text{allocated}$) = $8 - (1 + 1 + 2 + 2) = 2$

Need = max - allocation

Process	Need
P1	2
P2	1
P3	3
P4	4

P1 first to get resources: ($\text{resources} - \text{P1 need} \geq 0$), 0 resources left over

P1 finishes and releases resources (3), leaving 3 resources available in total

P2 is next to get resources (1), 2 resources now left over

P2 finishes and releases resources (2), leaving 4 resources available in total

P3 is next to get resources (3), 1 resource now left over

P3 finishes and releases resources (5), leaving 6 resources available in total

P4 is next to get resources (4), 2 resources now left over

P4 finishes and releases resources (6), leaving 8 resources available in total

Safe sequence: <P1, P2, P3, P4>

b.

If P4 were to request 1 resource, should the request be granted? (Yes or No)

If yes –then indicate a safe sequence from the resulting state

If No –then indicate the sequence of steps that would lead to a deadlock

Yes, P4's request would be granted, the sequence can already accommodate P4. If its cost were less, it wouldn't be a problem. Safe sequence would still be <P1, P2, P3, P4>.

2

Assume that the following section of main memory is used to store the page table for 3 different processes. The page-table base register values for process P1 is 1080, for P2 is 1085, and for P3 is 1090. Assume that the contents of memory below correspond to frame numbers. Also assume that frame size is 4096.

contents	3	5	8	4	7	2	0	15	11	18	6	20	24	9	10	13	30	38	40	1
Memory	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
addresses	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	7	7	7	8	8	8	8	8	8	8	8	8	8	9	9	9	9	9	9
	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

To which physical memory address would the logical address (1, 1200) correspond to if generated by P1?

Frame size = page size = 4096

P1 base = 1080

Segment 1 => $1080 + 1 = 1081$ => Frame number 2 => $4096 * 2 = 8192$

Offset = 1200

Physical address = $8192 + 1200 = 9392$

To which physical memory address would the logical address (3, 800) correspond to if generated by P2?

P2 base = 1085

Segment 3 => $1085 + 3 = 1088$ => Frame number 24 => $4096 * 24 = 98304$

Offset = 800

Physical address = $98304 + 800 = 99104$

Assume that each process has 5 pages. To which process does the following physical address belong and which logical address corresponds to each physical address:

Physical address 57344

Integer division of $57344 // 4096 = 14 = \text{frame number}$

Frame number 14 not present in page table; page fault occurs

Physical address 57343

Integer division of $57343 // 4096 = 13 = \text{frame number} \Rightarrow \text{address } 1091$

Process P3, logical address (1, 4095)

3

A paging system is experiencing a page fault rate of 1 in 1 million page references.

- Page fault occurs
 - 70% of time no frame replacement is needed
 - 10 ms to service
 - 30% of time a frame replacement is needed
 - 40% of this time the frame is modified, 20 ms to service
 - 60% of this time the frame is not modified, 12 ms to service
- It takes 100 nanoseconds to reference a physical memory location
 - Page table access time negligible

Calculate the effective memory access time for this system under these conditions. Clearly indicate intermediate steps.

$EAT = (1 - p) \times \text{memory access} + p (\text{page fault overhead} + \text{swap page out} + \text{swap page in} + \text{restart overhead})$

Memory access time: 100 ns

70% of the time: 10 ms service

12% of the time: 20 ms service

18% of the time: 12 ms service

$$EAT = (1 - P) \times 100ns + [(0.7P) \times 10ms] + [(0.12P) \times 20ms] + [(0.18P) \times 12ms]$$

$$P = 1/1000000$$

$$100ns = 1 \times 10^{-7} \text{ seconds}$$

$$10ms = 0.01 \text{ seconds}$$

$$EAT = 0.0000000999999s + 0.000000007s + 0.0000000024s + 0.00000000216s = 0.0000001115599s$$

$$0.0000001115599 \text{ s} = 111.59 \text{ ns}$$